

Review Seminararbeit

Seminar Software Engineering für verteilte Systeme



WiSe/SoSe 2024

Hinweise

- Länge des Reviews: 2–3 Seiten (inklusive dieser Hinweise)
- Jede Frage in diesem Template muss beantwortet werden! **Ersetzen** Sie dazu die vorhandenen `\todo`-Befehle im Template durch Ihre Antworten in Fließtext oder ausführlichen Stichpunkten.
- Die Qualität der von Ihnen verfassten Reviews geht in Ihre Gesamtnote für das Seminar ein.
- Entfernen Sie nach dem vollständigen Bearbeiten des Reviews das `todonotes`-Paket im Header dieser Datei. Wenn Sie alle `\todo`-Befehle ersetzt haben, kompiliert das Dokument weiterhin ohne Fehler. `\usepackage[color=smdsblue!25]{todonotes}`

Allgemeine Informationen

Titel der zu bewertenden Arbeit

Recent Approaches to accelerate CP Solving

Hauptinhalt der Arbeit

Constraint Programming findet Verwendung bei Optimierungsproblemen mit Nebenbedingungen. Zu diesen zählen: Vehicle Routing, Scheduling, Planung und vieles mehr. Besonders die Industrie steht vor einigen Scheduling-Problemen, bei denen Zeiteffizienz eine große Rolle spielt. Ein bekanntes kombinatorisches Problem ist das 3-SAT-Problem, das als boolesche Formel in konjunktiver Normalform vorliegt.

Es existieren bereits verschiedene Toolsets von Firmen wie IBM und Google. Constraint Satisfaction Problems (CSP) und Constraint Optimization Problems (COP) sind die zwei Arten von Constraint-Problemen. Während CSP irgendeine Lösung suchen, streben COP dagegen eine optimale Lösung des Problems an. COP können als Erweiterung der CSP gesehen werden. Das 3-SAT-Problem kann als Beispiel für einen CSP genommen werden. Eine Möglichkeit, um CSP zu lösen, ist mittels Backtracking, jedoch ist das Lösen eines solchen Problems NP-vollständig.

COP sind besonders interessant, denn in der Realität wird nicht nur nach einer Lösung, sondern nach optimalen Lösungen für bestimmte Probleme gesucht. Das Tripel der CSP wird um eine Zielfunktion erweitert, die minimiert oder maximiert werden soll. Je nach Problemstellung werden unterschiedliche Algorithmen verwendet, z.B. der Simplex-Algorithmus bei linearen Constraints und

Zielfunktion. Der Branch-and-Bound-Algorithmus wird bei Integer Programming Problems verwendet. Probleme, die sich nicht über lineare Zusammenhänge beschreiben lassen, können möglicherweise mit Gradientenverfahren gelöst werden, bei denen konvexe Funktionen verwendet werden und die nichtlineare Funktion durch eine konkave Zielfunktion approximiert wird.

Es werden insgesamt vier Ansätze zur Beschleunigung von Constraint Programming vorgestellt. Als erstes der Portfolio-Ansatz, bei dem mehrere Solver kombiniert werden, um die Leistung zu verbessern; die Wahl der Solver spielt eine wichtige Rolle. Mittlerweile werden Solver oft automatisch für die gegebenen Probleme ausgewählt, z.B. durch Entscheidungsbäume oder neuronale Netze. Eine weitere Methode ist das Model-Based Optimization, das verwendet wird, wenn die objektive Funktion nicht bekannt oder die Auswertung des Blackbox-Modells teuer ist. Bei diesem Ansatz wird anhand von Auswertungen ein Modell der Blackbox erstellt und dieses für die Optimierung verwendet. Mit der Auswertung der Blackbox wird ein Ergebnis bestimmt, das als Endresultat ausgegeben oder zur Verfeinerung des Modells verwendet wird.

Das dritte Verfahren ist Automated Parameter Tuning, bei dem die Parameter der Solver automatisch angepasst werden. Ein Konfigurator übergibt verschiedene Konfigurationen dem Zielalgorithmus anhand der Parameter und Domänen. Eine Kostenfunktion gibt an, wie gut der Zielalgorithmus konfiguriert ist, und die Parameter werden iterativ daran angepasst. Durch Grid Search oder Random Search wird nach den optimalen Parametern gesucht. Grid Search bietet den Vorteil, dass relevante Bereiche nicht ausgelassen werden, ist jedoch im Vergleich zu Random Search ineffizient, bei dem relevante Parameterkombinationen zufällig getestet werden.

Der letzte Ansatz ist eine Kombination aus CP und SAT. Mittels Lazy Clause Generation werden die Constraints eines CP-Problems in SAT-Klauseln umgewandelt, die den Vorteil haben, dass sie durch SAT-Solver schneller gelöst werden können.

Die Leistungsfähigkeit von CP-Systemen wird durch Forschung und Entwicklung weiter verbessert und die Anwendung auf reale Probleme erweitert. Die vier vorgestellten Methoden sind nur ein Teil der vorhandenen Ansätze zur Beschleunigung von Constraint Programming. In der Zukunft können neue Innovationen durch andere Technologien wie Quantum Computing zustande kommen, wodurch die Optimierung von CP-Systemen ein aktives Gebiet der Forschung mit großem Potenzial bleibt.

Allgemeine Bewertung

Stärken der Arbeit

Die Einführung bietet nicht nur eine grundlegende Erklärung der wesentlichen Konzepte des Constraint Programming (CP), sondern auch eine Verknüpfung mit relevanten mathematischen Modellen, wie beispielsweise beim 3-SAT-Problem in der Einleitung sowie bei Constraint Satisfaction Problems (CSP) und Constraint Optimization Problems (COP) in den Grundlagen.

Diese Erläuterungen schaffen eine solide Basis für das Verständnis der nachfolgenden Ansätze, die im weiteren Verlauf der Arbeit behandelt werden. Zudem werden Abbildungen sinnvoll eingesetzt, um die Thematik visuell darzustellen. Diese stehen in gutem Zusammenhang mit dem Text und unterstützen die Leser dabei, die komplexen Konzepte besser zu verstehen, wie beispielsweise bei der Erklärung der COP, der Verwendung von neuronalen Netzen und dem Parameter-Tuning-Prozess sowie den Suchalgorithmen, die dabei verwendet werden.

Insgesamt trägt die gut strukturierte Einleitung, zusammen mit den klaren mathematischen Definitionen und unterstützenden Abbildungen, wesentlich zur Verständlichkeit und Nachvollziehbarkeit der Arbeit bei. Dies ermöglicht es den Lesern, sich intensiv mit den vorgestellten Ansätzen auseinanderzusetzen und deren Anwendungspotenzial besser zu erfassen.

Schwächen der Arbeit

Bei der Erklärung der COP wird das Gradientenverfahren zur Lösung von nichtlinear zusammenhängenden Problemen als Möglichkeit vorgestellt. Leider wird jedoch nicht weiter auf die Funktionsweise dieses Verfahrens eingegangen.

Der Portfolio-Solver besteht, wie der Name schon sagt, aus einem Portfolio von Solvern. Es wurde jedoch kein spezifischer Solver genannt, der in diesem Portfolio enthalten sein könnte. Dies würde das Verständnis für die Leistungsfähigkeit des Ansatzes verbessern, da es klarer wäre, in welchen Situationen welcher Solver sinnvoll ist und welche Vorteile sie bieten.

Der Ansatz des Automated Parameter Tuning verwendet zwei Suchalgorithmen. Bei der Random Search werden nur relevante Parameterkombinationen auf zufällige Weise getestet. Es ist jedoch nicht klar, wie dieser Algorithmus erkennt, welche Kombinationen von Parametern relevant sind oder nicht.

Bei der letzten Methode, der Kombination von CP und SAT, wäre es interessant zu erfahren, wie genau die Constraints eines CP-Problems in SAT-Klauseln umgewandelt werden und in welchen Fällen die Umwandlung möglich oder schwierig ist.

Nutzung KI-basierter Tools

Einige Inhalte dieser Arbeit wurden mit Hilfe von KI-basierter Software generiert, was in der eidesstattlichen Erklärung festgehalten wird. Daher kann man davon ausgehen, dass einige Passagen dieser Arbeit von KI-Tools auf Grammatik überprüft wurden. Der Text lässt sich flüssig lesen, was ebenfalls auf eine mögliche Verbesserung durch KI hindeuten könnte. Allerdings ist anzumerken, dass sich einige Tippfehler im Text finden, was darauf schließen lässt, dass nicht jeder Teil des Textes von KI überprüft oder generiert wurde.

Sachliche Korrektheit

Das Literaturverzeichnis enthält viele Quellen, die Links zu den jeweiligen Programmen bzw. Firmen sind. Diese kommen hauptsächlich in der Einleitung vor und stellen keine große Informationsweiterung dar. Die genannten Quellen [1] sowie [8] bis [14] sind lediglich Links zu den Projekten oder deren Dokumentation. Bei der Zitierung von Quelle [7] wurden die Variablen n und m vertauscht. Im Originaltext steht n für die Orte und m für die Fahrzeuge. Diese Verwechslung kann zu Verwirrung führen, auch wenn sie möglicherweise beabsichtigt war.

Einige Seitenangaben findet man im Literaturverzeichnis anstatt im Text. Da jedoch manche Seiten im Text angegeben werden, wäre es aus Gründen der Konsistenz sinnvoll, sich auf eine Variante festzulegen.

Im Hauptteil der Arbeit werden auch Quellen zitiert, die nicht unbedingt wissenschaftlich sind, beispielsweise [25], eine PowerPoint-Präsentation bei der Erklärung von Automated Parameter Tuning. Des Weiteren wurde bei der Kombination von CP und SAT ein Link zu einer Webseite als Quelle verwendet.

Die Quellen, die überprüft wurden, sind korrekt zitiert, aber nicht alle Quellen konnten überprüft werden, da die genaue Ausgabe der Werke nicht gefunden werden konnte. Trotzdem konnte man Parallelen erkennen, auch wenn es sich nicht um die gleiche Version gehandelt hat.

Äußere Form

Grammatikalisch gibt es keine Fehler, die ich finden konnte. Wie bereits erwähnt, gibt es jedoch zwei Tippfehler, die mir aufgefallen sind: Auf Seite 1 "generationäusnutztin der unteren Hälfte der

Einleitung und auf Seite 6 'DDie' am Anfang des Schlussteils der Arbeit. Die Anzahl der Rechtschreib- und Grammatikfehler ist sehr gering.

Zu Beginn werden die Grundlagen erklärt, darunter die Begriffe Constraint Satisfaction Problem und Constraint Optimization Problem. Anschließend werden die vier Ansätze zur Beschleunigung von CP-Solving erläutert. Zum Schluss werden alle Ansätze noch einmal zusammengefasst und kurz auf die zukünftige Entwicklung eingegangen. Somit ist ein klarer roter Faden sowie eine sinnvolle Einteilung der Kapitel erkennbar.

Der Einsatz von Abbildungen ist angemessen. Es wurden weder zu viele noch zu wenige eingefügt, und sie sind zudem sinnvoll beschriftet. Eine direkte Referenz zu den Abbildungen im Text findet man zwar nicht, aber die Stellen, an denen sie eingefügt wurden, ergänzen den Text gut durch die Visualisierung.

Optisch gibt es an der Arbeit nichts auszusetzen. Sowohl die Abbildungen als auch die Formeln wurden so eingefügt, dass sie weder den Lesefluss stören noch untergehen.